Revisit 1D Total Variation restoration problem with new real-time algorithms for signal and hyper-parameter estimations

Zhanhao Liu, Marion Perrodin, Thomas Chambrion, Radu S. Stoica

Revisit 1D Total Variation restoration problem with new real-time algorithms for signal and hyper-parameter estimations

Zhanhao Liu[1,3,*], Marion Perrodin[3], Thomas Chambrion[2] and Radu S.Stoica[1]

December 17, 2020

Abstract

1D Total Variation (TV) denoising, considering the data fidelity and the Total Variation (TV) regularization, proposes a good restored signal preserving shape edges. The main issue is how to choose the weight λ balancing those two terms. In practice, this parameter is selected by assessing a list of candidates (e.g. cross validation), which is inappropriate for the real time application. In this work, we revisit 1D Total Variation restoration algorithm proposed by Tibshirani and Taylor. A heuristic method is integrated for estimating a good choice of λ based on the extremums number of restored signal. We propose an offline version of restoration algorithm in $O(n \log n)$ as well as its online implementation in $O(n)$. Combining the rapid algorithm and the automatic choice of λ, we propose a real-time automatic denoising algorithm, providing a large application fields. The simulations show that our proposition of λ has a similar performance as the states of the art.

1 Introduction

In this article, we consider the denoising of 1D raw signal. Such signals are mathematically modelled with u, a real function defined on a bounded open subset Ω of $\mathbb{R}$: $u : \Omega \to \mathbb{R}$. Finite number of noised samples of u are available: the noised observations $y = (y_1, \cdots, y_n)$ are sampled at $t = (t_1, \cdots, t_n)$ with $t_1 < \cdots < t_n$ and y_i the sample at t_i. We assume the sampling period $\tau_i = t_i - t_{i-1}$ is not constant, and the observation y_i is $u(t_i)$ adding an independent random noise ϵ:

$$y_i = u(t_i) + \epsilon \tag{1}$$

with $\mathbb{E}(\epsilon) = 0$ and $\mathbb{V}(\epsilon) = \sigma^2$.

1.1 Signal restoration methods

Our objective is to restore u by knowing y, which is also called *inverse problem*. We hope to have an automatic denoising method in a real time context with high precision and limited calculation resource demand. It is an active research domain, and we give here a short review of existing methods:

- Digital filter: for 1D signals with a constant sampling frequency, digital filters are widely used. For example, Savitzky–Golay filter [1] is one of the most popular. It consists in fitting a local polynomial regression to each sample point, and it can be formalized as a convolution with a fixed kernel. The method is extremely rapid, since the kernel can be pre-calculated. However, for the inconstant sampling period, the convolution kernel needs to be recalculated for each sample point, which makes this method inefficient.

* Contact: zhanhao.liu@univ-lorraine.fr

1 Université de Lorraine, CNRS, Inria, IECL, F-54000 Nancy, France

2 Institut de Mathématiques de Bourgogne, UMR 5584, CNRS, UBFC, F-21000 Dijon, France.

3 Saint-Gobain Research Paris, 39 Quai Lucien Lefranc, F-93300 Aubervilliers, France.

- Probabilistic approach: by introducing priors regarding the signal properties and the model parameters, the restoration is obtained by maximising the a posteriori distribution. See more details in [2].

- Variational denoising methods: it consists in at first defining an energy function $F(y, u) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and estimating the restored signal by:

$$u^* = \arg\min_u F(y, u) \tag{2}$$

  Usually, the energy function is a weighted sum of two terms:

$$F(y, u) = L(y, u) + \lambda D(u) \tag{3}$$

  with the *data fidelity* $L(y, u)$, the *regularization* $D(u)$ and a hyper-parameter $\lambda \in \mathbb{R}^+$ balancing the weight of two terms.

In this article, we focus on variational denoising methods with *Total Variation* (TV) regularization:

$$D(u) = \int_\Omega |\nabla u| \tag{4}$$

where ∇u is the gradient of u.

Total Variation based restoration method is first proposed in [3]. The authors in [4] propose an unconstrained minimisation problem:

$$u^*_{TV} = \arg\min_u \int_\Omega \lambda |\nabla u| + |u - y|^2 \tag{5}$$

for a given Lagrange multiplier $\lambda > 0$, and prove the uniqueness of solution as well as an one-to-one correspondence between the noise's variance σ^2 and λ. TV-based methods preserve sharp edges and propose a locally smooth restoration. It has been applied to signal ([5]), image ([3], [4]) and more generally to graph structure ([6], [7]).

For 1D signal, we can estimate efficiently the exact solution with a given weight parameter λ. We present here two families:

- Dynamic of λ ([8], [9]): those algorithms estimate the solution by following the dynamic of u^*_{TV} in function of λ. The authors in [8] propose a *path* algorithm to estimate Λ, the list of λ provoking a change of u^*_{TV} topology, by varying λ from $+\infty$ to 0. Those methods are efficient for estimating the solutions with several candidate values of λ.

- Dynamic of "new" sample: for a sequence of n points, the author in [10] proposes to add the samples one by one into consideration and update the solution sequentially based on the local behavior of TV-restoration [11]. This method is particularly appropriate for the online processing of a stream of data.

The performance of the restoration depends on the choice of λ, which is one of the obstacles for the real application. The automatic hyper-parameters selection is an active research domain for mathematics (probabilistic and variational approaches) and computer science. Traditional approaches consist in calculating a goodness-of-fit criterion for several candidate values and selecting the best one, including cross-validation. Those approaches choose randomly a part of data to test the performance of the model fitted without this part. It is difficult to find out the optimal λ in the real time context. Another approach is to introduce some prior knowledge, e.g. in knowing the largest number possible of constant piece of u, a good choice of λ can be proposed quickly [8]. Many efficient methods are based on the prior knowledge about the noise ϵ: the authors in [12] and [13] propose to select the parameter under Morozov's discrepancy principle [14]: e.g. the L^2 norm of restoration residuals $(y - u^*_{TV})$ is equal to the variance of noise, and

the authors in [15] propose to select the parameter based on the statistical characteristics of restoration residuals. If necessary, we can estimate the noise variance σ^2 by [16] and [17].

We present here more in detail two methods based on known σ^2 that work well for n points 1D signal:

- Stein unbiased risk estimation (SURE) [18] is an unbiased estimator of restoration error $|u^* - u_{net}|_2^2$ with u_{net} the original signal. [8] shows that the freedom degree of 1D TV-based restoration is the segment (c.f. Section 2.1) number of restored signal, noted K, which give us:

$$\text{SURE}(\lambda) = |y - u^*(\lambda)|_2^2 + 2\sigma^2 K - n\sigma^2 \tag{6}$$

  By assessing a list of candidates, we select the parameter λ giving the minimum of $\text{SURE}(\lambda)$:

$$\lambda_{\text{SURE}} = \arg\min \text{SURE}(\lambda) \tag{7}$$

- Adaptive universal threshold (AUT) [19] selects the parameter λ based on the behavior of the restoration with an elegant pre-choice of λ: $\lambda_N = \sigma/2\sqrt{n \log\log n}$. Based on the restoration with λ_N, we get an estimation of the number of segments $\hat{K}$. Thereafter, the authors adjust the choice of λ by:

$$\lambda_{AUT} = \frac{\sigma}{2}\sqrt{\frac{n}{\hat{K}} \log\log \frac{n}{\hat{K}}} \tag{8}$$

In addition, some heuristic methods are available: the authors in [9] propose to estimate λ by tracking the quantity $|u^*_{TV}(\lambda_l) - u^*_{TV}(\lambda_{l-1})|^2$ with λ_l the smallest λ giving a restoration with $l-1$ segments. More generally for the weight parameter of (3), some heuristic methods (e.g. L-curve [20]) are proposed based on the variation of $L(y,u)$ in function of $D(u)$.

1.2 Contributions

Our contributions are resumed as follows:

- First, by revisiting the algorithm proposed in [8] and [9], we propose an online approach to estimate the list Λ based on the local influence of a new sample for the TV-based method. This approach is efficient for both hyper-parameter selection and data stream restoration.
- Second, we propose an automatic choice of the weight parameter λ based on the numbers of the restored signal's local extremums in function of λ without any prior knowledge on the signal.

The overall time complexity (for both the automatic choice of the weight parameter λ and the reconstruction of the signal) is in $O(n)$ for online implementation with a space complexity in $O(n)$.

1.3 Structure of this paper

The structure of this work is as follow: in Section 2, we will at first revisit the results of [8] and [9] about 1D TV-restoration method and analyse the influence of the introduction of a new sample point. Then, based on the established theoretical elements, we propose (Section 3) respectively some (offline and online) algorithms for estimating Λ, u^*_{TV} with a given λ and a good choice of hyper-parameter λ for a good performance of restoration. Thereafter (Section 4), we compare our method with different existing methods on some simulated and measured signals. Finally, conclusions and perspectives are depicted. For the sake of readability, the proofs are gathered in Appendix A.

2 Theoretical analysis: revisit of 1D Total Variation Restoration

We aim to recover the unknown vector $u = (u_1, \cdots, u_n)$ from the noisy observations $y = (y_1, \cdots, y_n)$ with y_i the sample at time t_i. We introduce the sampling period vector $\tau = (\tau_1, \cdots, \tau_n)$ with:

$$\tau_i = \begin{cases} t_i - t_{i-1}, & i = 2, ..., n. \\ t_2 - t_1, & i = 1. \end{cases} \tag{9}$$

We consider the minimization of the discrete approximation of (5):

$$\begin{aligned} F(\cdot, y, \tau, \lambda) :& \mathbb{R}^n \to \mathbb{R} \\ & u \mapsto L(y, u, \tau) + \lambda D(u) \end{aligned} \tag{10}$$

with data fidelity term $L(y, u, \tau) = \sum_{i=1}^{n} \tau_i (y_i - u_i)^2$ and total variation regularization term $D(u) = \sum_{i=1}^{n-1} |u_i - u_{i-1}|$. For the sake of simplicity, $F(u, y, \tau, \lambda)$ is noted as $F_n(\lambda)$ for the case of n sample points.

The restored signal is given by:

$$u^*(\lambda) = (u_1^*(\lambda), \cdots, u_n^*(\lambda)) = \arg\min F_n(\lambda) \tag{11}$$

By following, we will at first introduce the segment notation of the restoration proposed in [9]. Then, we will analyse the dynamic of $u^*(\lambda)$ in function of λ and the local modification of the restoration due to the induction of a new sample at the end of sequence.

2.1 Segment notation

The considered functional (10) is convex but not derivable. Since we work on a finite sample of signal, the solution $u^*(\lambda)$ can be seen as piece-wise constant. We use a similar notation as [9] to represent the constant piece: a set of index $\{j, j+1, \cdots, k\}$ of consecutive points whose restored value $u_j^*(\lambda) = \cdots = u_k^*(\lambda)$ is called a *segment* if it can not be enlarged, which means if $u_{j-1}^*(\lambda) \neq u_j^*(\lambda)$ (or $j = 1$) and $u_k^*(\lambda) \neq u_{k+1}^*(\lambda)$ (or $k = n$). The number of segments of $u^*(\lambda)$ is noted as $K(\lambda)$. The following notations are introduced for the j^{th} segment of $u^*(\lambda)$:

- Index set $\mathcal{N}_j(\lambda) = \{i_1^j, \cdots, i_{n_j}^j\}$ with $n_j(\lambda) = \text{Card}(\mathcal{N}_j(\lambda))$, containing the point index inside the j^{th} segment.
- Segment level $v_j^*(\lambda) = u_i^*(\lambda), \forall i \in \mathcal{N}_j(\lambda)$

An equivalent representation of $u^*(\lambda)$ is provided by the *set of segment levels* $v^*(\lambda) = (v_1^*(\lambda), \cdots, v_{K(\lambda)}^*(\lambda))$, with $v_1^*(\lambda) \neq v_2^*(\lambda)$, $v_2^*(\lambda) \neq v_3^*(\lambda)$, $\cdots$, $v_{K(\lambda)-1}^*(\lambda) \neq v_{K(\lambda)}^*(\lambda)$, and the *cutting set* (or signal structure) $\mathcal{N}(\lambda) = \{\mathcal{N}_1(\lambda), \cdots, \mathcal{N}_{K(\lambda)}(\lambda)\}$. The total variation under the segment representation is given by (12) which is derivable.

$$TV(v(\lambda)) = \sum_{j=1}^{K(\lambda)-1} |v_{j+1}(\lambda) - v_j(\lambda)| \tag{12}$$

With a given cutting set $\mathcal{N}(\lambda)$, $v^*(\lambda)$ can be estimated by:

$$\begin{aligned} v^*(\lambda) &= \{v_1^*(\lambda), \cdots, v_K^*(\lambda)\} \\ &= \arg\min\{\sum_{j=1}^{K(\lambda)} \sum_{i \in \mathcal{N}_j} \tau_i (y_i - v_j)^2 + \lambda \sum_{j=1}^{K(\lambda)-1} |v_{j+1} - v_j|\} \end{aligned} \tag{13}$$

Let $s(v) = \{s_0(v), s_1(v), \cdots, s_K(v)\}$ with

$$s_i(v) = \begin{cases} 0 & i = 0 \text{ or } K \\ \text{sign}(v_{i+1} - v_i) & i = 1, \cdots, K-1 \end{cases} \tag{14}$$

$K = \text{Card}(v)$ and $s^* = s(v^*(\lambda))$, first order optimality conditions of (13) imply:

$$v_j^*(\lambda) = \overline{y}_j^* + \frac{\lambda}{2\mathcal{T}_j}(s_j^* - s_{j-1}^*) \tag{15}$$

with $j = 1, \cdots, K(\lambda)$, the segment length $\mathcal{T}_j = \sum_{i \in \mathcal{N}_j(\lambda)} \tau_i$ and the mean value inside j^{th} segment $\overline{y}_j^* = \frac{\sum_{i \in \mathcal{N}_j(\lambda)} \tau_i y_i}{\mathcal{T}_j}$.

We introduce some items for facilitating the presentation, illustrated in Figure 1:

- We will call the local maximum segment *max segment*, the local minimum segment *min segment* and the rest *neutral segment*.
- We will call the last point of a segment (excepted the last segment) the *junction* of segments.

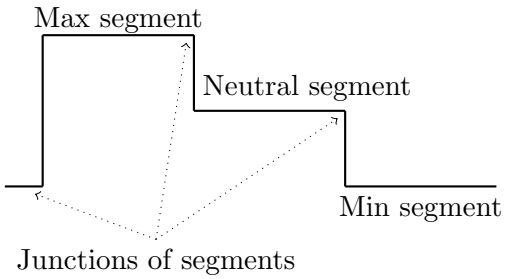


Figure 1: Illustration of max, neutral, min segments and the junction of segments. The last point of the last segment is not a junction of segments

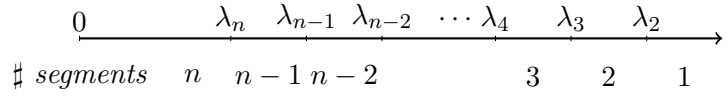
2.2 Influence of hyper-parameter λ on cutting set $\mathcal{N}(\lambda)$

When the cutting set $\mathcal{N}(\lambda)$ is known, the computation of $v^*(\lambda)$ is direct by (15). In this section, we present the influence of λ on the cutting set $\mathcal{N}(\lambda)$.

The authors in [8] and [9] describe the behavior of solution $u^*(\lambda)$ as λ decreases. Inspired from an elegant theorem in [21] and [2] (Proposition 2.5.1, p52) about the piece-wise relation between the restoration with Potts model and the parameter λ, we reformulate the results in [8] by the following theorem:

Theorem 1. *There is a sequence* $\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n, \lambda_{n+1})$ *with* $0 = \lambda_{n+1} \leq \lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_2 < \lambda_1 = \infty$ *such that* $\forall \lambda_a, \lambda_b \in [\lambda_{l+1}, \lambda_l)$ *with* $l = 1, \cdots, n$, *we have the same cutting set* $\mathcal{N}^*(\lambda_a) = \mathcal{N}^*(\lambda_b)$.

Remark 1. *In the continuous setting, only two segments are merged for* $\lambda \in \Lambda$. *We have* $0 = \lambda_{n+1} < \lambda_n < \lambda_{n-1} < \cdots < \lambda_2 < \lambda_1 = \infty$ *such that* $\forall \lambda \in [\lambda_{l+1}, \lambda_l)$ *with* $l = 1, \cdots, n$, *the restored signal* $u^*(\lambda)$ *has* l *segments: Card*$(\mathcal{N}^*(\lambda)) = l$.



2.3 Dynamic of segments level in function of λ

Theorem 1 allows us to break the estimation of Λ into some simpler sub-problems: λ_l can be estimated from λ_{l+1} for $1 < l \leq n$. We will describe the dynamic of the restored signal $u^*(\lambda)$ while λ increases.

For $\lambda = 0$, (11) is the least-square estimation, providing $u^*(0) = y$. Assuming that $y_i \neq y_{i+1}$ for every i, $u^*(\lambda)$ has n segments for $0 \leq \lambda < \lambda_n$, each segment having only one point.

For simplicity of the presentation, we assume that only two segments are merged for $\lambda \in \Lambda$. Following Remark 1, Λ has $n+1$ distinct elements with $\lambda_{n+1} = 0$ and $\lambda_1 = +\infty$.

As λ increases, (15) implies that the min and max segments approach to their neighbors, while the neutral segments remain the same level. For $\lambda \in \Lambda$, two neighbor segments have the same level and form together a new segment. Let's take an example: we assume λ_{l+1} known which gives a solution with l segments. Let this solution be $v^l(\lambda) = \{v_1^l(\lambda), \cdots, v_l^l(\lambda)\}$ and $\mathcal{N}^l(\lambda) = \{\mathcal{N}_1^l(\lambda), \cdots, \mathcal{N}_l^l(\lambda)\}$, we introduce also the following notations:

$$\beta_j^l = \frac{1}{2\mathcal{T}_j^l}(s_j^l - s_{j-1}^l) \tag{16}$$

$$\Gamma^l = (\gamma_1^l, \cdots, \gamma_{l-1}^l) = (\beta_1^l - \beta_2^l, \cdots, \beta_{l-1}^l - \beta_l^l) \tag{17}$$

$$\Delta v^l(\lambda) = (v_1^l(\lambda) - v_2^l(\lambda), \cdots, v_{l-1}^l(\lambda) - v_l^l(\lambda)) \tag{18}$$

with $s^l = s(v^l)$ and $\mathcal{T}_j^l = \sum_{i \in \mathcal{N}_j^l} \tau_i$.

Let $\lambda_{l+1} \leq \lambda_a < \lambda_b < \lambda_l$, since the cutting set stays the same, we have:

$$\Delta v_i^l(\lambda_a) = \Delta v_i^l(\lambda_b) + \gamma_i^l(\lambda_a - \lambda_b) \tag{19}$$

and $|\Delta v_i^l(\lambda_a)| \geq |\Delta v_i^l(\lambda_b)|$ for every i. The cutting set changes when $\Delta v_i^l(\lambda) = 0$: two segments are merged together. The merged segment index is given by:

$$k = \arg\min_k(|\Delta v_k^l(\lambda_{l+1})/\gamma_k^l|) \tag{20}$$

λ_l can be obtained by:

$$\lambda_l = \lambda_{l+1} + |\Delta v_k^l(\lambda_{l+1})/\gamma_k^l| \tag{21}$$

The value of λ_l provoking this merge depends on the nature (i.e. min, max or neutral) of those two neighbors (s_{k-1}^l, s_k^l and s_{k+1}^l), their length ($\mathcal{T}_k^l$ and $\mathcal{T}_{k+1}^l$) and the difference between their segment levels (Δv_k^l).

The solution with $l-1$ segments, $v^*(\lambda_l)$ and $\mathcal{N}(\lambda_l)$, is given by the following equations:

$$v^*(\lambda_l) = \begin{cases} v_j^l + \beta_j^l(\lambda_{l+1} - \lambda_l), & j = 1, \cdots, k-1 \\ v_{j+1}^l + \beta_{j+1}^l(\lambda_{l+1} - \lambda_l), & j = k, \cdots, l-1 \end{cases} \tag{22}$$

$$\mathcal{N}(\lambda_l) = \begin{cases} \mathcal{N}_j^l, & j = 1, \cdots, k-1 \\ \{\mathcal{N}_k^l, \mathcal{N}_{k+1}^l\}, & j = k \\ \mathcal{N}_{j+1}^l, & j = k+1, \cdots, l-1 \end{cases} \tag{23}$$

When $\lambda > \lambda_2$, the total variation regularization becomes too important, and all the points form one single segment.

In summary, when λ increases, the difference between two neighbour segments levels (i.e. $|v_j^* - v_{j+1}^*|$) becomes smaller, and the total variation of the restored signal decreases.

2.4 Influence of hyper-parameter λ on extremums number of restored signal

A good choice of λ provides a restoration preserving the intrinsic variation of signal and eliminating the local oscillation introduced by noise at the same time. However, the segment number $\mathrm{Card}(\mathcal{N}^*(\lambda))$ may not be a good indicator of the local oscillation of the restored signal since the neutral segments do not contribute to total variation of restored signal.

Let's return to (12): only the max and mix segments are considered in the total variation. We note $g(\lambda)$ the number of extremums of the restored signal $u^*(\lambda)$. A noised signal has a large value of total variation and a large number of local extremums due to the local oscillation, while a well-restored signal which keeps (ideally) only the intrinsic variation of signal may have a small $g(\lambda)$. $g(\lambda)$ seems to us a good indicator of $D(u^*(\lambda))$ and allows us to estimate a correct choice of λ for the restoration.

In this section, we will analyse the relation between $g(\lambda)$ and λ. Theorem 2 shows the monotonicity of the extremums number $(g(\lambda))$ in function of λ.

Theorem 2. *There is a sequence $\Lambda^g = (\lambda_1^g, \lambda_2^g, \cdots) \subset \Lambda$ with $\infty = \lambda_0^g > \lambda_1^g > \lambda_2^g > \cdots$ such that:*

- *$g(\lambda_a) = g(\lambda_b)$, $\forall \lambda_a, \lambda_b \in [\lambda_{l+1}^g, \lambda_l^g)$ for every l.*
- *$g(\lambda') > g(\lambda'')$, $\forall \lambda' \in [\lambda_{l+1}^g, \lambda_l^g)$ and $\forall \lambda'' \in [\lambda_l^g, \lambda_{l-1}^g)$ for every l.*

$g(\lambda)$ can be estimated directly from Λ, shown in Proposition 1.

Proposition 1. *Assumes that only two segments are merged together for $\lambda_l \in \Lambda$. In this case, at least one of the two merged segments is a local extremum. Let $\Delta g(\lambda_l) = g(\lambda_{l+1}) - g(\lambda_l)$:*

- *If only one segment is an extremum, then the new segment is also an extremum. $\Delta g(\lambda_l) = 0$.*
- *If both segments are extremums and neither are the first or the last segment, the new segment is not an extremum. $\Delta g(\lambda_l) = -2$.*
- *If both segments are extremums and one is the first or the last segment, then the new segment is an extremum. $\Delta g(\lambda_l) = -1$.*

We propose a simple method to calculate $\Delta g(\lambda_l)$: let $\{v^l, \mathcal{N}^l\}$ be the solution for $\lambda = \lambda_{l+1}$, $\mathcal{N}_j^l$ and $\mathcal{N}_{j+1}^l$ be the two segments to merge, and $s^l = s(v^l)$ following (14), $\Delta g(\lambda_l)$ can be obtained following Table 1.

Conditions		Results
$s_{j-1}^l \times s_{j+1}^l$	$s_{j-1}^l + s_j^l + s_{j-1}^l$	$\Delta g(\lambda_l)$
$\neq 0$		$= -\lvert s_{j-1}^l + s_{j+1}^l \rvert$
$= 0$	< 2	$= -1$
	$= 2$	$= 0$

Table 1: Calculation of $\Delta g(\lambda_l)$ in function of $s_{j-1}^l \times s_{j+1}^l$ and $s_{j-1}^l + s_j^l + s_{j-1}^l$ with $\mathcal{N}_j^l$ and $\mathcal{N}_{j+1}^l$ the two segments to merge.

2.5 Local diffusion of a new observation

For 1D signal, the new samples arrive sequentially. Assume n samples are collected and the new sample (y_{n+1}, t_{n+1}) arrive with $t_n < t_{n+1}$. In this section, we show that the restoration change locally with the introduction of the new sample for a fixed λ.

Let $u^* = (u_1^*, \cdots, u_n^*) = \arg\min F_n$ based on the n first observations with λ fixed and $\hat{u} = (\hat{u}_1, \cdots, \hat{u}_n, \hat{u}_{n+1}) = \arg\min F_{n+1}$ with $F_{n+1} = F_n + (t_{n+1} - t_n)(y_{n+1} - u_{n+1})^2 + \lambda|u_{n+1} - u_n|$, we introduce the following notations:

- For $\hat{u} = \arg\min F_{n+1}$:
  - Segment level set: $\hat{v} = \{\hat{v}_1, \cdots, \hat{v}_{\hat{K}}\}$.
  - Index set of j^{th} segment: $\hat{\mathcal{N}}_j = \{\hat{i}_1^j, \cdots, \hat{i}_{\hat{n}_j}^j\}$ with $\hat{n}_j = \text{Card}(\hat{\mathcal{N}}_j)$.
  - Segment length set: $\hat{\mathcal{T}} = \{\hat{\mathcal{T}}_1, \cdots, \hat{\mathcal{T}}_{\hat{K}}\}$ with $\hat{\mathcal{T}}_j = \sum_{i \in \hat{\mathcal{N}}_j} \tau_i$
- For $u^* = \arg\min F_n$:
  - Segment level set: $v^* = \{v_1^*, \cdots, v_{K^*}^*\}$.
  - Index set of j^{th} segment: $\mathcal{N}_j^* = \{i_1^{j,*}, \cdots, i_{n_j^*}^{j,*}\}$ with $n_j^* = \text{Card}(\mathcal{N}_j^*)$.
  - Segment length set: $\mathcal{T}^* = \{\mathcal{T}_1^*, \cdots, \mathcal{T}_{K^*}^*\}$ with $\mathcal{T}_j^* = \sum_{i \in \mathcal{N}_j^*} \tau_i$

We can establish the following theorem for the influence of y_{n+1} over the restoration of n first samples:

Theorem 3. *If there exists an index $j \in \{2, \cdots, K^*(\lambda)\}$ such that $sign(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = sign(v_{K^*(\lambda)}^*(\lambda) - y_{n+1})$, then the new reconstruction $\hat{u}$ satisfies $\hat{u}_i(\lambda) = u_i^*(\lambda)$ for all $i < i_1^{j,*}$.*

The diffusion from y_{n+1} changes only the last part of u^* up to the junction of two segments whose sign of the variation $v_{j-1}^*(\lambda) - v_j^*(\lambda)$ corresponds to that of $v_{K^*}^*(\lambda) - y_{n+1}$, costless to detect. For updating $u^*(\lambda)$ with y_{n+1}, only the last points are necessary, and we will introduce the following definition:

Definition 1. *A sequence $(u_m^*, \cdots, u_n^*)$ is called* non-isolated *with $m = i_1^{j*}$ where j is the last segment which satisfies $sign(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = sign(v_{K^*}^*(\lambda) - y_{n+1})$.*

Theorem 4 shows the length of the non-isolated sequence decreases in function of λ. For $\lambda = 0$, each segment contains only one point, and the new sample creates a new segment without any influence to the first n points. For a large value of λ, all the points form a segment giving the global mean value, so all the restored signal points are influenced by the new sample.

Theorem 4. *Let $\lambda_1 < \lambda_2$ and the length of the non-isolated sequence of $u^*(\lambda)$ be $l(\lambda)$, we have $l(\lambda_1) \leq l(\lambda_2)$.*

In summary, the new sample (y_{n+1}, t_{n+1}) changes only the last part of the restoration $u^*(\lambda)$ with a given λ, and the influence of the new sample diffuses "further" with the augmentation of λ.

2.6 Independence between segments of restored signal

In this section, we show that the merges of points inside a segment are independent of the points outside the segment.

We propose to save Λ in a new vector $\Lambda^\circ = (\lambda_1^\circ, \lambda_2^\circ, \cdots, \lambda_{n-1}^\circ)$ with λ_i° the value of λ for which the points i and $i+1$ are merged into the same segment. For each segment of the restored signal $u^*(\hat{\lambda})$ with a given $\hat{\lambda}$, we add one point at the junction of segments by following the definition below:

Definition 2. *Let* $\hat{\lambda}$ *and* $\epsilon_\lambda > 0$*, we have* $v^*(\hat{\lambda}) = \{v_1^*(\hat{\lambda}), \cdots, v_l^*(\hat{\lambda})\}$*,* $\mathcal{N}^*(\hat{\lambda}) = \{\mathcal{N}_1, \cdots, \mathcal{N}_l\}$*,* $l = K(\hat{\lambda})$ *and* $s_i = sign(v_{i+1}^*(\hat{\lambda}) - v_i^*(\hat{\lambda}))$*. For each segment* $\{y_{\mathcal{N}_j}, \tau_{\mathcal{N}_j}\}$ *with* $j = 1, \cdots, l$*, let* $c_i = \frac{\hat{\lambda}+\epsilon_\lambda}{2s_i}$*, we introduce the* virtual segment $\{y^+_{\mathcal{N}_j}, \tau^+_{\mathcal{N}_j}\}$ *where:*

$$y^+_{\mathcal{N}_i} = \begin{cases} \{y_{\mathcal{N}_1}, v_1^*(\hat{\lambda}) + c_1\}, & i = 1. \\ \{v_i^*(\hat{\lambda}) - c_{i-1}, y_{\mathcal{N}_i}, v_i^*(\hat{\lambda}) + c_i\}, & i = 2, ..., l-1. \\ \{v_l^*(\hat{\lambda}) - c_{l-1}, y_{\mathcal{N}_l}\}, & i = l. \end{cases} \tag{24}$$

$$\tau^+_{\mathcal{N}_i} = \begin{cases} \{\tau_{\mathcal{N}_1}, 1\}, & i = 1. \\ \{1, \tau_{\mathcal{N}_i}, 1\}, & i = 2, ..., l-1. \\ \{1, \tau_{\mathcal{N}_l}\}, & i = l. \end{cases} \tag{25}$$

Following the dynamic of the restoration described in Section 2.3, the variation of segment level depends on the sign of Δv but not its value. For a segment of $u^*(\hat{\lambda})$ (i.e j^{th} segment $\mathcal{N}_j^*(\hat{\lambda}) = \{i_1^{j*}, \cdots, i_1^{(j+1)*} - 1\}$), the only influences from the points outside the segment are the signs of Δv on the borders of the segment (i.e. for the first point of segment $\text{sign}(u^*_{i_1^{j*}}(\hat{\lambda}) - u^*_{i_1^{j*}-1}(\hat{\lambda}))$ and for the last point of segment $\text{sign}(u^*_{i_1^{(j+1)*}}(\hat{\lambda}) - u^*_{i_1^{(j+1)*}-1}(\hat{\lambda}))$). Under the same border conditions, the merges of sub-segments inside a segment of $u^*(\hat{\lambda})$ are independent of the points outside the segment, which will merge with this segment for $\lambda > \hat{\lambda}$. For each virtual segment, those border conditions are guaranteed by the introduction of points at the junction of segments. In consequence, we can break the estimation of Λ° into some independent sub-problems for each virtual segment, shown in the following proposition:

Proposition 2. *Let* Λ° *the estimation with all the samples* $\{y_i, t_i\}_{1,\cdots,n}$*,* $\Lambda_i^\circ = \{\lambda_{i,1}^\circ, \cdots\}$ *the estimation with* i^{th} *virtual segment* $(y^+_{\mathcal{N}_i}, \tau^+_{\mathcal{N}_i})$ *and*

$$\Lambda_i^* = \begin{cases} \{\lambda_{1,1}^\circ, \cdots, \lambda_{1,n_1-1}^\circ\}, & i = 1. \\ \{\lambda_{i,2}^\circ, \cdots, \lambda_{i,n_i}^\circ\}, & i = 2, ..., l. \end{cases} \tag{26}$$

we have $\cup_{i=1,\cdots,l}\Lambda_i^* = \{\lambda | \lambda \leq \hat{\lambda} \cap \lambda \in \Lambda^\circ\}$.

Proposition 2 allows us to estimate separately Λ° for each segment of $u^*(\hat{\lambda})$ except the junctions of segments since they have $\lambda^\circ > \hat{\lambda}$. Combining with the local influence of the new sample to the restoration (i.e $u^*(\hat{\lambda})$), the independence between segments implies that the introduction of new sample brings a local modification of Λ°, respectively the non-isolated sequence of $u^*(\hat{\lambda})$ and the junction of $u^*(\hat{\lambda})$'s segments. The local influence of the new sample points motivates the online estimation of Λ°.

3 Proposition of Algorithms

In this section, we will propose some real-time algorithms for estimating a good choice of λ and the restoration $u^*(\lambda)$ with a given λ.

3.1 Estimation of the elements of Λ and their corresponding solution

The author in [8] established a *path* algorithm for estimating Λ by moving the parameter from $\lambda = \infty$ to $\lambda = 0$. In the following, we will revisit this algorithm in order to get simultaneously Λ and $g(\lambda)$.

Following the dynamic described in Section 2.3, the estimation of an element of Λ consists in finding two segments to merge by (20) and updating all segments in function of λ by (22) and (23). But the update of the non-merged segments is useless: since β does not change, the variation of the segment level v in function of λ remains linear with the same slope (β). For estimating λ_l giving the solution with $l-1$ segments, the most important is the minimum of $|\frac{\Delta v^l}{\Gamma^l}|$ providing the value of λ_l and the position of the merge. The merge of two segments changes up to two elements of Γ^l, respectively the left and right neighbors of the new merged segment.

Let $\eta = (\eta_1, \cdots, \eta_{n-1})$ with:

$$\eta_i = \frac{y_{i+1} - y_i}{\beta_{i+1} - \beta_i} \tag{27}$$

and $\beta = (\beta_1, \cdots, \beta_n)$ where β is obtained following (16) with y and τ, we introduce the order of merge $n^\circ = \arg\operatorname{sort}(\eta, \text{ascending} = \text{True})$. At each iteration, we need to treat the first element of n°, and reproduce n° for the remaining segments after changing the value of η for the new segment's neighbor(s).

Besides, at each merge of segments, we save two new values $(\lambda_{new}, \Delta g(\lambda_{new}))$, respectively λ_{new} the λ value provoking the merge and $\Delta g(\lambda_{new})$ the variation of $g(\lambda)$ after the merge. We propose to save those new values in two vectors of size $n-1$ by following Section 2.6:

- $\Lambda^\circ = (\lambda_1^\circ, \lambda_2^\circ, \cdots, \lambda_{n-1}^\circ)$ with λ_i° the value of λ for which the points i and $i+1$ are merged into the same segment.
- $\Delta g^\circ = (\Delta g(\lambda_1^\circ), \Delta g(\lambda_2^\circ), \cdots, \Delta g(\lambda_{n-1}^\circ))$

The algorithm (DP-TV) is shown in Algorithm 1. At each iteration, after finding two merged segments by the first element of n°, the new values are saved at the junction of two merged segments (the last point of the left merged segment).

Remark 2.
1. *With a careful implementation of the computation of n° (i.e. binary search tree), it can be implemented in low time and space complexity, respectively $O(n \log n)$ and $O(n)$.*
2. *If the original noised signal has constant pieces (i.e. $\exists i, y_i = y_{i+1}$), we need to replace the constant piece $\{y, \tau\}_{j,\cdots,k}$ by $\{y, \sum_{i=j}^k \tau_i\}$ before applying our algorithm. Under the assumption of continuous noise, the probability to have a constant piece is equal to 0, but it remains a practical issue of implementation in the real data with actual floating point arithmetic.*
3. *A major practical issue is the merge of multiple segments for the same value of λ due to the errors inherent to floating point arithmetic of the micro-processor. With a slight modification, DP-TV can deal with this situation, in which the first elements of η are equal.*

3.2 Estimation of $u^*(\lambda)$ and $g(\lambda)$

With Λ° estimated by Algorithm 1, we need to apply the following method for estimating the solution $u^*(\lambda)$ with a given λ:

- Find the cutting set: $j = \{j_1, \cdots, j_{l+1}\}$ with $j_1 = 0$, $j_{l+1} = n$ and $\lambda_{j_i}^\circ > \lambda$ for $1 < i \leq l$.
- Initialisation: v_{output} and $\mathcal{T}_{output}$ two vectors of size l. For each segment $[j_i + 1, j_{i+1}]$ with $1 \leq i \leq l$: $\mathcal{T}_{output,i} = \sum_{m=j_i+1}^{j_{i+1}} \tau_m$ and $v_{output,i} = \frac{1}{\mathcal{T}_{output,i}} \sum_{m=j_i+1}^{j_{i+1}} \tau_m y_m$.

Algorithm 1 DP-TV: Estimation of all elements of Λ and their corresponding solution

Require: $y = (y_1, \cdots, y_n),\ \tau = (\tau_1, \cdots, \tau_n)$
  $\tilde{v} = y,\ \tilde{\tau} = \tau$
  $\tilde{\lambda} = (0, \cdots, 0)$ ▷ Initialisation
  $s = s(\tilde{v})$ following (14)
  $\tilde{\beta} = \beta$ following (16) with $\tilde{\tau}$ and s
  Left neighbour vector $nl = (0, \cdots, n-1)$
  Right neighbour vector $nr = (2, \cdots, n+1)$
  Get $\tilde{\Gamma}$ and $\Delta\tilde{v}$ by (17) and (18) with β and y
  $\eta = |\Delta\tilde{v}/\tilde{\Gamma}|$
  $n^\circ = \arg\operatorname{sort}(\eta, \text{ascending} = \text{True})$
  for $i = 1, \cdots, n-1$ **do**
    $k = n^\circ_i$ ▷ Find segments to merge
    $v_{new} = \tilde{v}_k + \tilde{\beta}_k(\eta_k - \tilde{\lambda}_k)$
    $\lambda_{new} = \eta_k$
    $\tau_{new} = \tilde{\tau}_k + \tilde{\tau}_{nr_k}$
    $\Lambda^\circ_k = \lambda_{new}$ ▷ Save for output
    Get Δg°_k by Proposition 1
    $k_{left}, k_{right} = nl_k, nr_k$ ▷ Update η for next merge
    $\tilde{\lambda}_{k_{right}}, \tilde{v}_{k_{right}}, \tilde{\tau}_{k_{right}} = \lambda_{new}, v_{new}, \tau_{new}$
    $\tilde{\beta}_{k_{right}} = \frac{1}{2\tau_{new}}(s_{k_{right}} - s_{k_{left}})$
    $nl_{k_{right}}, nr_{k_{left}} = k_{left}, k_{right}$
    $k_{rr} = nr_{k_{right}}$
    if $k_{left} >= 1$ **then** ▷ Not first segment
      $v_{left} = \tilde{v}_{k_{left}} + (\lambda_{new} - \tilde{\lambda}_{k_{left}})\tilde{\beta}_{k_{left}}$
      $\eta_{k_{left}} = |\frac{v_{left} - v_{new}}{\tilde{\beta}_{k_{left}} - \tilde{\beta}_{k_{right}}}| + \lambda_{new}$
    end if
    if $k_{right} < n$ **then** ▷ Not last segment
      $v_{rr} = \tilde{v}_{k_{rr}} + (\lambda_{new} - \tilde{\lambda}_{k_{rr}})\tilde{\beta}_{k_{rr}}$
      $\eta_{k_{right}} = |\frac{v_{new} - v_{rr}}{\tilde{\beta}_{k_{right}} - \tilde{\beta}_{k_{rr}}}| + \lambda_{new}$
    end if
    Resort $n^\circ_{\{i+1,\cdots,n-1\}}$ following $\eta_{n^\circ_{\{i+1,\cdots,n-1\}}}$
  end for
  return $\Lambda^\circ, \Delta g^\circ$

- Adjust the segment levels with the given λ: the level of i^{th} segment is given by:

$$v_i^*(\lambda) = v_{output,i} + \beta_i \lambda \tag{28}$$

with $\beta_i = \frac{1}{2\mathcal{T}_{output,i}}(s_i - s_{i-1})$ and $s = s(v_{output})$.

The complexity for estimating $u^*(\lambda)$ with a given λ is in $O(n)$.

The estimation of $g(\lambda)$ is given by:

$$g(\lambda) = 1 - \sum_{i=1}^{n-1} \Delta g_i^\circ \mathbf{1}_{\{\lambda_i^\circ - \lambda\}} \tag{29}$$

with:

$$\mathbf{1}_\alpha = \begin{cases} 1, & \alpha > 0 \\ 0 & \alpha \leq 0 \end{cases} \tag{30}$$

The complexity for estimating $g(\lambda)$ is in $O(n \log n)$ since we need to sort Λ°. We can reduce it to $O(n)$ by saving directly Λ and $\Delta g(\lambda)$ for $\lambda \in \Lambda$ at each iteration, instead of Δg°.

3.3 Estimation of λ from $g(\lambda)$

What is a good λ? Our objective is to denoise the observations y. By knowing the true signal u_{net}, the mean squared error of the reconstruction $u^*(\lambda)$ is defined by:

$$\mathcal{R}(\lambda) = \frac{1}{n} ||u_{net} - u^*(\lambda)||_2^2 \tag{31}$$

A good λ must have a small value of $\mathcal{R}(\lambda)$. The optimal λ can be found out by:

$$\lambda_{op} = \arg\min \mathcal{R}(\lambda) \tag{32}$$

Usually, only some prior knowledge or even nothing about u_{net} is known. Here, we propose a rapid deterministic approach to find out a similar value of λ_{op} without any prior knowledge about the original signal u_{net} and the noise ϵ. The only assumption we made is following: the noise ϵ represents the high frequency of the observed signal y, while the original signal u_{net} represents the low frequency. This assumption is natural for the signal processing community.

With a known cutting set, the denoising is simply done by an average over the points inside each segment. A good denoising cutting must regroup enough points in each segment. The value of λ need to be chosen carefully:

- A small λ provides a fine cutting: averaging locally over a small portion of points limits the denoising effect.
- A large λ gives a coarse cutting: the structure of u is vanished by the global average over a big portion of points, and it provides a poor denoising performance.

The question is how to get a good compromise between local and global behaviors. The key feature is the number of min-max segments $g(\lambda)$ of $u^*(\lambda)$. Following (15), the segment level v is piece-wise linear in function of λ, and $\frac{\partial v_j^*}{\partial \lambda}$ depends on segment length $\mathcal{T}_j$: short segment is more sensitive to the variation of λ. Supposing a moderate noise is added to the original signal, we can have the following observations:

- For a small λ, the restored signal has a huge number of small min-max segments introduced by noise. A small augmentation of λ can merge two min-max segments.
- For a large λ, we have some long segments due to the intrinsic structure of signal. We need a large increase of λ to merge two min-max segments.

In consequence, the small values of λ (named *denoising regime*) vanish the extremums introduced by noise in averaging over a fine cutting, and the large values of λ (named *destructing regime*) regroup the intrinsic extremums of signal. The above analysis shows a specific structure of $g(\lambda)$: $g(\lambda)$ decreases quickly in denoising regime, but slowly in destruction regime. In the transitory region between two regimes, the noises are nearly all removed, but the shape of original signal is preserved. To sum up, a good λ may correspond to the discontinuity of the *tendency* (or gradient) of $g(\lambda)$'s decreasing.

We present a simulation of block signal [22] illustrated in Figure 2a: y is a piece-wise constant signal (noted u_{net}) adding a Gaussian noise $\epsilon \sim \mathcal{N}(0,1)$. $g(\lambda)$ of this signal is shown in Figure 2b. The shape of $g(\lambda)$ corresponds well to our analysis:

- Denoising regime: for $\lambda < 2$, $g(\lambda)$ decreases quickly since some small min-max segments are merged.
- Destructing regime: for $\lambda > 7$, $g(\lambda)$ decreases slowly. For $\lambda > 500$, the total variation regularisation becomes too important, and $g(\lambda)$ decreases to 1: $u^*(\lambda)$ has only one segment with $u^*(\lambda) = \text{mean}(y)$.

Our intuition is the following: a good λ must lie on the last part of transitory region between desnoising regime and the destructing regime, and the transitory region corresponds to the tendency discontinuity of $g(\lambda)$ in function of $\log(\lambda)$ behind a shape decreasing.

Since $g(\lambda)$ is monotonically decreasing and piece-wise constant, the estimation of the tendency of $g(\lambda)$ in function of $\log(\lambda)$ is not obvious. We use the following approximations: for $\lambda \in \Lambda^g$, we introduce the numerical differential operators:

$$\partial g(\lambda)^+ = g(q\lambda) - g(\lambda) \tag{33}$$

$$\partial g(\lambda)^- = g(\lambda) - g(\lambda/q) \tag{34}$$

with $q > 1$. $\partial g(\lambda)^-$ and $\partial g(\lambda)^+$ are respectively the approximation of the left and right derivative of $g(\lambda)$ in function of $\log(\lambda)$.

The approximation of second derivative of $g(\lambda)$ is given by:

$$\partial^2 g(\lambda) = \partial g(\lambda)^+ - \partial g(\lambda)^- \tag{35}$$

$\partial^2 g(\lambda)$ with $\log_{10}(q) = 1$ for every $\lambda \in \Lambda^g$ is shown in Figure 2c.

We propose the following method to estimate a good λ:

- Get $\partial^2 g(\lambda)$ by (35) for $\forall \lambda \in \Lambda^g$, and the transitory region is given by:

$$\lambda_{trans} = \arg\max \partial^2 g(\lambda) \tag{36}$$

- Adjustment: the transitory region has a similar value of $\partial^2 g(\lambda)$, and $\partial^2 g(\lambda)$ decreases sharply in destruction regime. A good estimation of λ can be given by the last $\lambda \in \Lambda^g$ of transitory region. One proposition is to find out the first sharp decreasing of $\partial^2 g(\lambda)$ for all $\lambda \in \{\lambda \in \Lambda^g\} \cap \{\lambda \geq \lambda_{trans}\}$. We propose to estimate the choice of λ by:

$$\lambda_{ours} = \arg \min_{\lambda \in \{\lambda \in \Lambda^g\} \cap \{\lambda \geq \lambda_{trans}\}} \partial^4 g(\lambda) \tag{37}$$

with:

$$\partial^4 g(\lambda_i^g) = \partial^2 g(\lambda_{i+2}^g) - 2\partial^2 g(\lambda_{i+1}^g) + \partial^2 g(\lambda_i^g) \tag{38}$$

The value of q needs to be large in order to avoid the local variation of the gradient of $g(\lambda)$. Typically, $0.5 \leq \log_{10}(q) \leq 1$ can well approximate the tendency of $g(\lambda)$. Besides, q can be chosen automatically: we propose q as the length of a long step of $g(\lambda)$ (in logarithm), and our proposition is $q = \max(\log_{10}(\lambda_{i+1}^g / \lambda_i^g))$ without the two first steps of $g(\lambda)$.

The complexity for estimating λ_{ours} from $g(\lambda)$ is in $O(n)$.

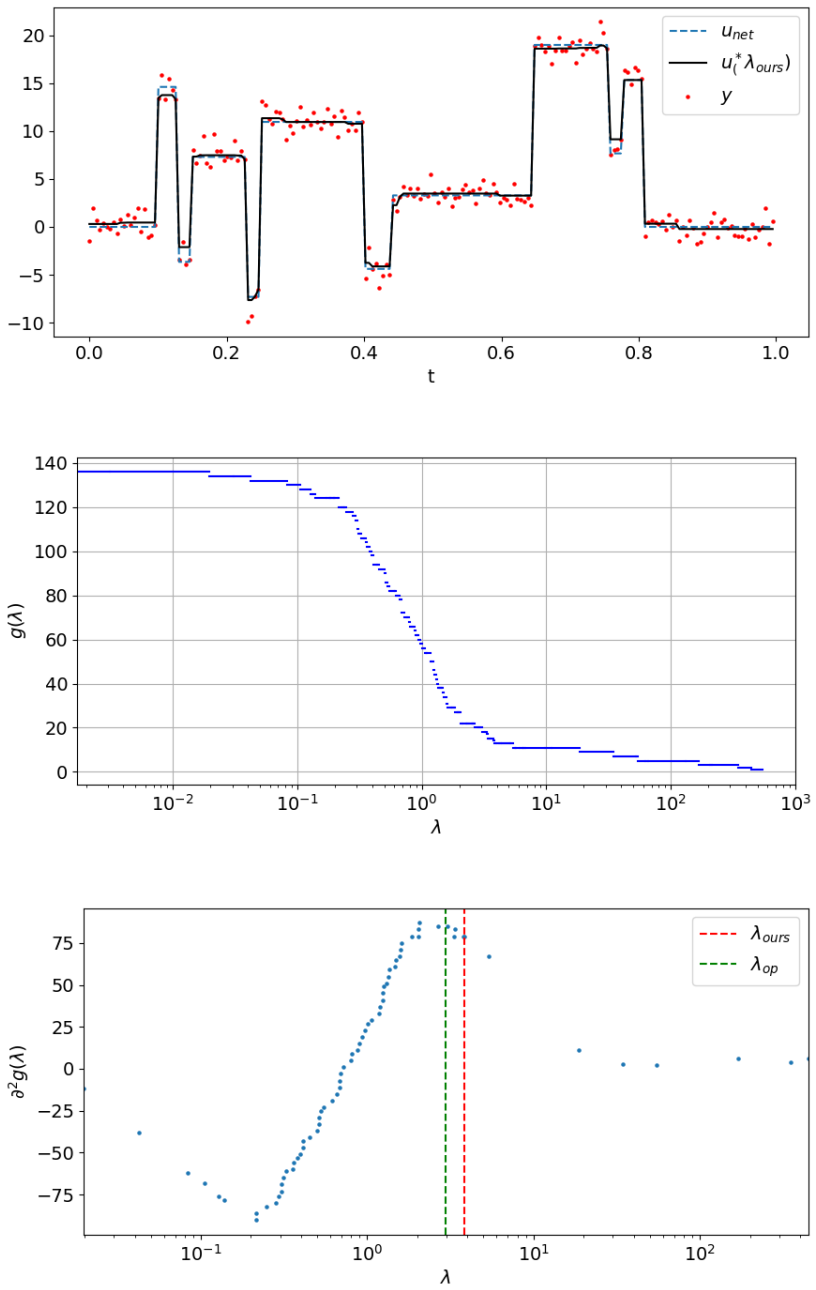


Figure 2: Simulation: how to choose a good λ from the extremums number ($g(\lambda)$). Top: Simulated block signal $y = u_{net} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$, SNR = 16.91dB, and the restoration with our proposition $u^*(\lambda_{ours})$ is shown in black; Middle: the extremums number ($g(\lambda)$) in function of λ; Bottom: $\partial^2 g(\lambda)$ with $\log_{10}(q) = 1$ for every $\lambda \in \Lambda^g$. Color: green ($\lambda_{op} = \arg\min \mathcal{R}(\lambda)$), red (our approach).

3.4 Online implementation of DP-TV

Our objective is to restore a huge amount of noised signals in a real time context. It asks for a weak temporal and spatial complexity for estimating the restoration $u^*(\lambda_{ours})$. In this section, based on the local modification introduced by the new sample, we will propose an online implementation of Algorithm 1.

We note $(\Lambda^{\circ,n}, \Delta g^{\circ,n})$, a set of two vectors of size $n-1$, the solutions of Algorithm 1 with n samples. With the new sample (y_{n+1}, t_{n+1}), the new results $(\Lambda^{\circ,n+1}, \Delta g^{\circ,n+1})$ based on

$n+1$ samples can be obtained by updating $(\Lambda^{\circ,n}, \Delta g^{\circ,n})$. We will only talk about the online estimation of $\Lambda^{\circ,n+1}$ from $\Lambda^{\circ,n}$ in detail. In the following, we note an application of Algorithm 1 to a given sequence of signal $(\{y\},\{\tau\})$ as $\Lambda^\circ = \text{DP-TV}(\{y\},\{\tau\})$.

After choosing a value of λ, noted $\hat{\lambda}$, we can get the restoration $u^*(\hat{\lambda})$ for the first n points. Following Theorem 3, the introduction of the new sample changes only the non-isolated sequence of $u^*(\hat{\lambda})$. Besides, Proposition 2 implies that $\lambda_i^{\circ,n} = \lambda_i^{\circ,n+1}$ if $\lambda_i^{\circ,n} \leq \hat{\lambda}$ for the isolated sequence, and the merge of the new restored signal $\hat{u}(\hat{\lambda})$'s segments (i.e $\{\lambda_i^{\circ,n+1} > \hat{\lambda}\}$) depends on the value of y_{n+1}.

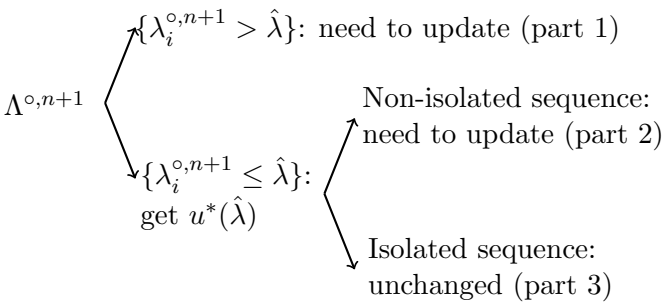


Figure 3: Illustration of online implementation: the changed and unchanged part of $\Lambda^{\circ,n+1}$ with the introduction of the new sample.

$\hat{\lambda}$ is indeed a cutting point of $\Lambda^{\circ,n+1}$: $\{\lambda^{\circ,n+1} \leq \hat{\lambda}\}$ and $\{\lambda^{\circ,n+1} > \hat{\lambda}\}$ will be treated separately. We note m and j^* respectively the first point and the first segment of the non-isolated sequence of $u^*(\hat{\lambda})$ following Definition 1. The results based on the first n points can be cut into three parts, illustrated in Figure 3, and we will detail the update procedure for each part.

We treat at first the unchanged part of $\Lambda^{\circ,n+1}$ (part 3 of Figure 3): all $\lambda_i^{\circ,n+1} \leq \hat{\lambda}$ remain the same for $i < m$. Formally, we have the following equation:

$$\lambda^{\circ,n+1}_{\{\lambda^{\circ,n+1}_{1,\cdots,m-1} \leq \hat{\lambda}\}} = \lambda^{\circ,n}_{\{\lambda^{\circ,n}_{1,\cdots,m-1} \leq \hat{\lambda}\}} \tag{39}$$

For the non-isolated sequence (part 2 of Figure 3), $\Lambda^a = \lambda^{\circ,n+1}_{\{\lambda^{\circ,n+1}_{m,\cdots,n} \leq \hat{\lambda}\}}$ can be estimated by $\text{DP-TV}(y^+_{\{m,\cdots,n+1\}}, \tau^+_{\{m,\cdots,n+1\}})$ where:

- $y^+_{\{m,\cdots,n+1\}} = \{v^*_{j^*}(\hat{\lambda}) - \frac{\hat{\lambda}+\epsilon_\lambda}{2\text{sign}(v_{j^*} - v_{j^*-1})}, y_{\{m,\cdots,n+1\}}\}$
- $\tau^+_{\{m,\cdots,n+1\}} = \{1, \tau_{\{m,\cdots,n+1\}}\}$

with $\epsilon_\lambda > 0$.

The non-isolated and isolated sequences can be assembled into $\Lambda^{temp} = \{\lambda_1^{temp}, \cdots, \lambda_n^{temp}\}$ with:

$$\lambda_i^{temp} = \begin{cases} \lambda_i^{\circ,n}, & i < m. \\ \lambda^a_{i-m+2}, & i \geq m. \end{cases} \tag{40}$$

It remains the update of $\{\lambda_i^{\circ,n+1} > \hat{\lambda}\} = \{\lambda_i^{temp} > \hat{\lambda}\}$ (part 1 of Figure 3). Let $b = \{\lambda^{\circ,n+1} > \hat{\lambda}\}$ containing indeed all the junctions of $\hat{u}(\hat{\lambda})$'s segments, we can get $\Lambda^b = \lambda_b^{\circ,n+1} = \text{DP-TV}(\hat{v}(\hat{\lambda}), \hat{\mathcal{T}}(\hat{\lambda}))$.

Finally, $\Lambda^{\circ,n+1}$ can be assembled in the following way:

$$\lambda_i^{\circ,n+1} = \begin{cases} \lambda_i^{temp}, & \text{if } \lambda_i^{temp} \leq \hat{\lambda}. \\ \lambda^b_{p(i)} & \text{if } \lambda_i^{temp} > \hat{\lambda}. \end{cases} \tag{41}$$

with $p : \mathbb{Z} \to \mathbb{Z}$ giving the index of i in the vector b.

The other solution vector $\Delta g^{\circ,n+1}$ can be estimated from $\Delta g^{\circ,n}$ in the same way as $\Lambda^{\circ,n+1}$ (from $\Lambda^{\circ,n}$).

The online version of DP-TV is summed up in Algorithm 2: DP-TV is applied two times on a small part of the data, and the results are merged following (41). Concerning the complexity, only a small part of $\Lambda^{\circ,n+1}$ needs to be recalculated: we obtain $u^*(\hat{\lambda})$ in $O(n)$, Λ^a in $O((n-m)\log(n-m))$ and Λ^b in $O(l_b \log l_b)$ with $l_b = \text{Card}(\Lambda^b)$. The overall complexity depends on the cutting point $\hat{\lambda}$, a trade-off between the complexity of Λ_a and Λ_b:

- A large value of $\hat{\lambda}$ makes u^* less fluctuating, and the non-isolated sequence may be long, even as long as y, which means $m = 1$. In this case, the computation of Λ_a is nearly $O(n \log n)$, the same as the offline implementation.
- The reconstruction with small value of $\hat{\lambda}$ has a little non-isolated sequence. However, all $\lambda > \hat{\lambda}$ with $\lambda \in \Lambda^{\circ,n}$, and need to be recalculated. In the worst case, all the elements of $\Lambda^{\circ,n}$ need to be recalculated in $O(n \log n)$

A good choice of $\hat{\lambda}$ may provide a short non-isolated sequence and a little size of Λ_b, which means $(n-m)\log(n-m) << n$ and $l_b \log(l_b) << n$. In this case, the complexity of the online implementation is $O(n)$. We propose to use the value estimated by the deterministic approach $\hat{\lambda} = \lambda_{ours}$ or slightly larger than λ_{ours} (i.e $\hat{\lambda} = 2\lambda_{ours}$). See more details in Section 4.2.

For estimating $g(\lambda)$ in $O(n)$, we propose to save $\Delta g(\lambda)$ following the order of Λ. For the online implementation, we believe that the computation of $\Delta g(\lambda)$ for $n+1$ points and Λ^{n+1}, the ordered list of $\Lambda^{\circ,n+1}$, can been seen as an insertion of a small ordered list (Λ of part 2 in Figure 3) into a large ordered list (Λ of part 3 in Figure 3). After insertion, we need to simply concatenate the new list with Λ of part 1 for the final result of Λ^{n+1}.

Algorithm 2 Online implementation of DP-TV

Require: $(y_1, \cdots, y_n), (\tau_1, \cdots, \tau_n),\ (y_{n+1}, \tau_{n+1})$
Require: $\Lambda^{\circ,n}, \Delta g^{\circ,n}, \hat{\lambda}$
Find non-isolated sequence $(m, \cdots, n)$ of $u^*(\hat{\lambda})$
$(\Lambda^a, v^a, \Delta g^a) = \text{DP-TV}(y^+_{\{m,\cdots,n+1\}}, \tau^+_{\{m,\cdots,n+1\}})$
$\Lambda^{\circ,n+1} = (\lambda_1^{\circ,n+1}, \cdots, \lambda_{n+1}^{\circ,n+1}) = \{\Lambda^{\circ,n}_{\{1,\cdots,m-1\}}, \Lambda^a_{\{2,\cdots\}}\}$
$\Delta g^{\circ,n+1} = \{\Delta g^{\circ,n}_{\{1,\cdots,m-1\}}, \Delta g^a_{\{2,\cdots\}}\}$
$b = \{\lambda^{\circ,n+1} > \hat{\lambda}\}$
$(\lambda_b^{\circ,n+1}, \Delta g_b^{\circ,n+1}) = \text{DP-TV}(\hat{u}(\hat{\lambda}), \hat{\mathcal{T}}(\hat{\lambda}))$
return $\Lambda^{\circ,n+1}$ and $\Delta g^{\circ,n+1}$

4 Applications

We have proposed an automatic TV-restoration method for the real time context with limited computation resource. In this section, we will evaluate the restoration performance and the execution time with some simulated signal, and show an application with some real data collected from a Saint-Gobain's plant.

4.1 Restoration performance evaluation

In the section, we evaluate the restoration with λ_{ours} proposed by our method (37) and compare with different existent methods.

4.1.1 Metrics and criteria

We use the mean square error $\mathcal{R}(\lambda)$ (31) between the restored signal and the original signal to evaluate a candidate of λ. The optimal value λ_{op} is given by (32). For comparing the performance between two given values λ_1 and λ_2, we apply the following criteria:

$$d(\lambda_1, \lambda_2) = \mathcal{R}(\lambda_1) - \mathcal{R}(\lambda_2) \tag{42}$$

$d(\lambda, \lambda_{op}) \geq 0, \forall \lambda \in \mathbb{R}^+$, since λ_{op} provides the minimum of $\mathcal{R}(\lambda)$. The smaller $d(\lambda, \lambda_{op})$ is, the better the estimation λ is.

However, it is not clear how to fix a threshold over $d(\lambda, \lambda_{op})$ between a good and unacceptable estimation of λ. It depends on the variance of noise and the shape of the original signal. An alternative way is to compare with the estimation with K-fold cross-validation.

4.1.2 Cross-validation

The objective of cross validation is to assess how a model behaves on an independent data set. By selecting the best model, we get an estimation of hyper-parameter. If $d(\lambda, \lambda_{op})$ is near $d(\lambda_{cv}, \lambda_{op})$ with λ_{cv} estimated by cross-validation and λ estimated by another method, we can say this method has a similar performance as cross-validation.

We show how we use cross-validation (CV) in 1D total variation denoising. The observed signal $\{y_i, t_i\}_{1,\cdots,n}$ is splitted randomly into k folds with (nearly) equal size in each fold, noted $(\mathbf{y}^i, \mathbf{t}^i)$ for the i^{th} fold and $m_i = \text{Card}(\mathbf{y}^i)$. A model is estimated without a fold of data. For example, the model without i^{th} fold is noted:

$$\mathbf{u}^{-i}(\lambda) = \arg\min_u \{F(\mathbf{y}^{-i}, u, \mathbf{t}^{-i}, \lambda)\} \tag{43}$$

where $(\mathbf{y}^{-i}, \mathbf{t}^{-i})$ the observed signal without i^{th} fold.

The prediction $\hat{u}^{-i}(t)$ at time t is given by the linear interpolation of $(\mathbf{u}^{-i}(\lambda), \mathbf{t}^{-i})$. The empirical error of $\mathbf{u}^{-i}(\lambda)$ is estimated over i^{th} fold which does not participate into the construction of the model:

$$e^{-i}(\lambda) = \frac{1}{m_i} \sum_{(y,t)\in(\mathbf{y}^i,\mathbf{t}^i)} (y - \hat{u}^{-i}(t))^2 \tag{44}$$

Finally, we can estimate the hyper-parameter λ by:

$$\lambda_{cv} = \arg\min_{\lambda\in\mathbb{R}^+} \sum_{i=1}^{K} e^{-i}(\lambda) \tag{45}$$

Remark 3. *A sequence of signal can be seen as a time series. The random split of training and validation sets is not appropriated for time series [23], since training and validation set are not independent anymore. However, we are interested in the performance of our model in training data, but not in an independent data set, so the random split of data is justified.*

4.1.3 Results

We are interested in 2 types of periodic signal (u_{net}) shown in Figure 4: piece-wise constant and piece-wise linear. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is added to u_{net}. Examples of simulated noised signals are also shown in Figure 4. Now, we compare our proposition, $\lambda = \lambda_{ours}$, estimated by (37) using $\log_{10} q \in \{0.5, 1\}$ and q automatic, with the proposition by 10-fold cross-validation (45). For each type of signal and each $\sigma \in \{0.5, 1, 1.5, 2, 2.5, 3\}$, 500 simulations are done.

The statistical characteristics of $d(\lambda_{ours}, \lambda_{op})$ with different parameters q and $d(\lambda_{cv}, \lambda_{op})$ are shown in Figure 5. For low noise variance, all those methods provide a reconstruction

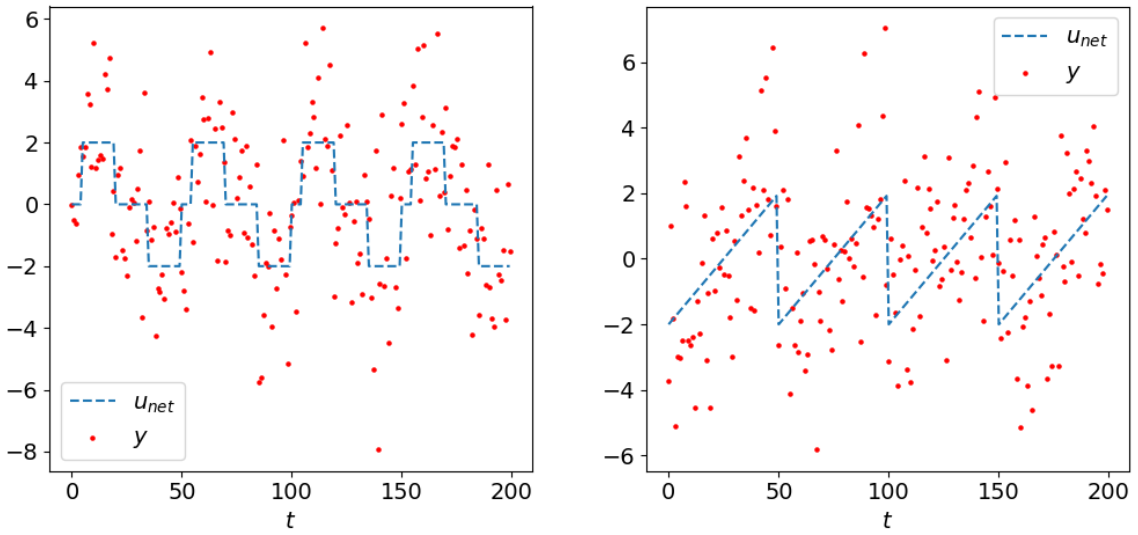


Figure 4: 2 types of simulated periodic signal (u_{net}). Examples of $y = u_{net} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 4)$ are shown. Left: Piece-wise constant; Right: Piece-wise linear.

$u^*(\lambda)$ nearly identical to $u^*(\lambda_{op})$. With the growing noise variance, the estimation of λ is more and more difficult, and the performance of all methods decreases. For high noise variance (i.e. $\sigma = 3$), those methods still provide a similar restoration as $u^*(\lambda_{op})$ for 50% cases with $d(\lambda, \lambda_{op}) < 0.1$.

Our methods with different parameters have a similar restoration performance as cross-validation for both types of periodic signal.

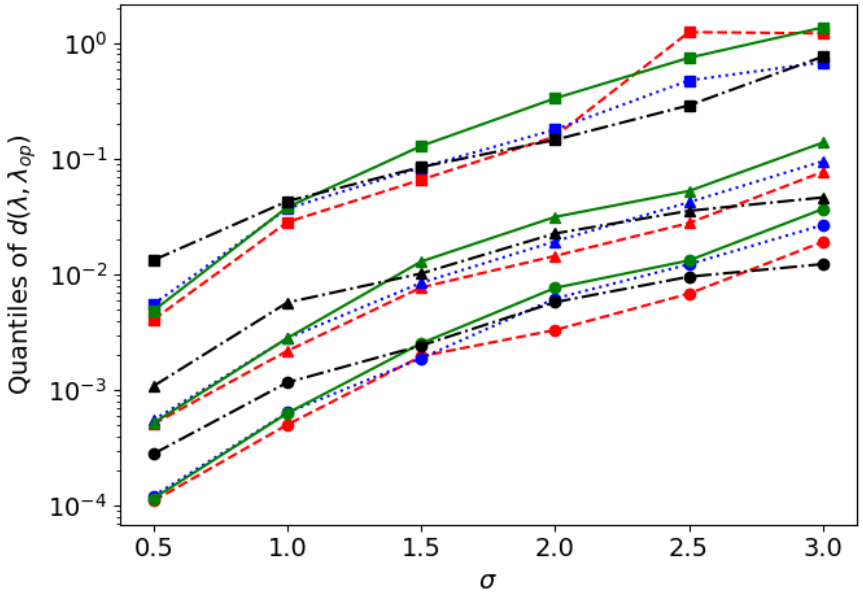
4.1.4 Comparison with existing methods

We compared our approach with some existing methods: Stein unbiased risk estimation (SURE) [18] and Adaptive universal threshold (AUT) [19]. Both methods are based on the known noise variance (σ^2). SURE assesses a criteria for several candidates λ, while AUT calculates $u^*(\lambda)$ for only two nice choices of λ. By following, we will compare those methods with σ known or unknown. In the case of unknown, the noise variance is estimated by [16].

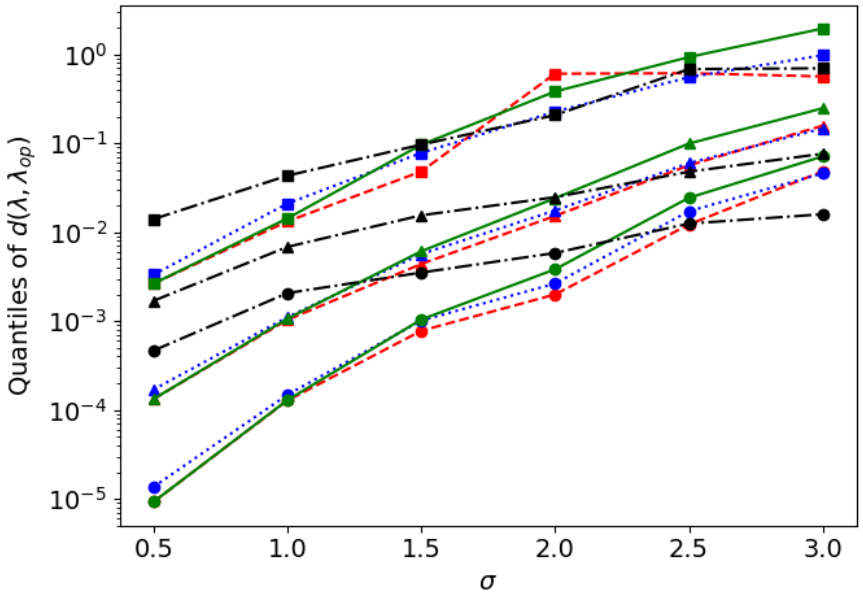
At first, we applied those methods to restore a non-periodic signal, shown in Figure 6, with different types of noise. The average value of $\mathcal{R}(\lambda)$ of different methods over 500 simulations are summarized in Table 2. The results show that our approaches with $q \in \{0.5, 0.75, 1\}$ and q automatic provide a similar performance as SURE for different types of noise, and out-perform AUT in the strong noise cases.

The second signal we tested is block signal (c.f Figure 2a) with different sampling points (frequency). The average values of $\mathcal{R}(\lambda)$ with $n = 199$, 499 and 999 are shown in Table 3. The performance of all the methods improves as the increasing of the sampling points. AUT and SURE have a smaller error than our approaches, but the difference between ours and the other approaches remains around the order of magnitude of 0.01. Indeed, the performance of our approaches based on the extremums number depends on the sampling frequency: average over a big number of points allows a better restoration of the intrinsic extremums of the original signal.

Concerning the complexity for estimating the choice of λ and the corresponding solution $u^*(\lambda)$, in applying a solution algorithm in $O(n \log n)$ (e.g. ours), SURE can estimate a choice of λ in $O(n \log n + Nn)$ with N the number of candidates and $O(Nn)$ the complexity for estimating $u^*(\lambda)$ from Λ° for all candidates, while AUT and our method are in $O(n \log n)$. For the online



(a) Piece-wise constant



(b) Piece-wise linear

Figure 5: $d(\lambda, \lambda_{op})$'s statistical characteristics of 500 simulations for 2 types of periodic signal with different Gaussian noise variance. We have tested our approach with $\log_{10}(q) = 0.5, 1$ and automatic choice of q. Solid line (green): automatic choice of q; Dots line (blue): $\log_{10}(q) = 0.5$; Dashed line (red): $\log_{10}(q) = 1$; Dash-dot line (black): 10-fold cross-validation. Circle marker: 25% quantile; Triangle marker: median; Square marker: 95% quantile.

implementation, the complexity of SURE is in $O(n + Nn)$, while AUT and our approach are in $O(n)$. In the case of huge number of candidates, SURE is slower than AUT and our method.

We can not compare our approach with the heuristic method proposed in [9] due to the lack of implementation details. This method tracks the variation of $|u^*(\lambda_{l-1}) - u^*(\lambda_l)|_2^2$ for every elements of Λ. However, the estimation of all $u^*(\lambda)$ for every $\lambda \in \Lambda$ is in $O(n^2)$, and this method

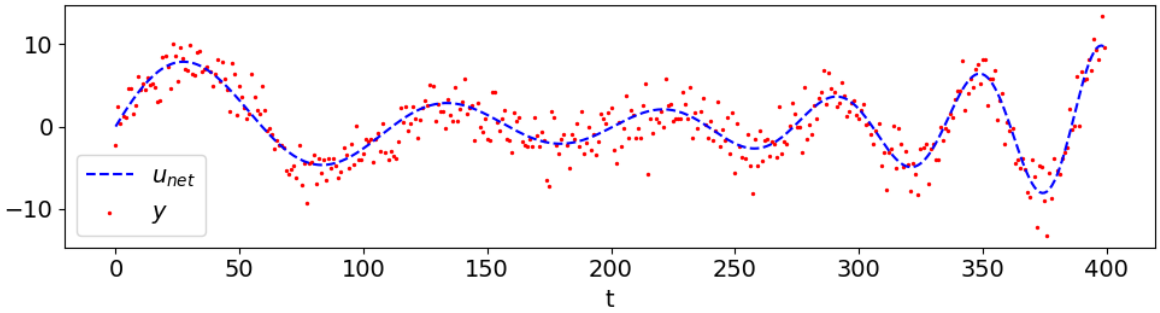


Figure 6: Example of non-periodic simulated signal. u_{net}: original signal; y: noised signal. $y = u_{net} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 4)$.

may not be appropriated for the real time context.

In summary, our approach works well for the signal restoration under different types of noise with high sampling frequency.

	$\mathcal{N}(0,1)$	$\mathcal{N}(0,2^2)$	$\mathcal{N}(0,3^2)$	$\mathcal{N}(0,2^2)$ $+\mathcal{U}[-2,2]$
$\min \mathcal{R}(\lambda)$	16.16	41.87	73.04	51.59
σ known				
AUT	16.43	58.87	160.28	88.31
SURE	17.07	45.31	81.67	56.39
σ unknown, $\hat{\sigma}$ estimated by [16]				
AUT	16.52	60.17	160.11	90.51
SURE	17.22	46.44	85.02	57.82
Our approach				
$\log_{10}(q) = 0.5$	16.64	44.60	82.44	55.38
$\log_{10}(q) = 0.75$	16.58	44.04	80.26	54.41
$\log_{10}(q) = 1$	16.58	44.30	83.95	55.67
q automatic	16.58	45.02	87.64	56.77

Table 2: The average value of $\mathcal{R}(\lambda) * 100$ of 500 simulations with a non-periodic signal shown in Figure 6 with different types of noise (Gaussian and uniform) shown in the first line of table.

4.2 Execution time evaluation

In this section, we compare the execution time of offline and online implementations. We measure the execution time for estimating Λ° with DP-TV (offline, $O(n \log n)$) and its online implementation. For the online implementation, the estimation with $n + 1$ points is based on the result with n points, while the offline implementations re-estimate from scratch the results. All the results shown are the average over 20 simulations: from 50 to 500 sample points of a periodic piece-wise linear whose one period is the same as Figure 4b.

The performance of the online implementation depends on the choice of the cutting point $\hat{\lambda}$. We will, at first, fix $\hat{\lambda} = 4$, and then discuss about the choice of $\hat{\lambda}$. The average execution time is shown in Figure 7a. Online implementation has a much smaller execution time than offline implementation: online implementation needs less than 2ms for the 500^{th} sampling point, while DP-TV needs more than 5ms for a sequence of 500 points.

We compare some choices of the cutting point $\hat{\lambda}$, and the results are shown in Figure 7b.

	199	499	999
$\min \mathcal{R}(\lambda)$	23.30	11.49	6.42
σ known			
AUT	25.01	11.92	6.56
SURE	24.97	12.24	6.83
σ unknown, $\hat{\sigma}$ estimated by [16]			
AUT	26.34	12.08	6.59
SURE	25.26	12.42	6.84
Our approach			
$\log_{10}(q) = 0.5$	27.88	13.36	7.75
$\log_{10}(q) = 0.75$	28.55	13.57	7.39
$\log_{10}(q) = 1$	30.17	14.63	7.72
q automatic	30.54	13.52	7.51

Table 3: The average value of $\mathcal{R}(\lambda) * 100$ of 500 simulations of block signal (SNR=16.91dB) with different sampling points. Examples of original and noised signal are shown in Figure 2a.

For too small and too large value of $\hat{\lambda}$ (i.e. $\hat{\lambda} = 0.1$ and 89.67), the execution time is nearly the same as the offline method since almost all the elements are included in Λ^b for the first case and in Λ^a for the second case, which need to be recalculated. The parameter $\hat{\lambda} \in \{1.64, 4, 18.44\}$, providing a small execution time, seems to us a good choice of the cutting point: all needs less than 3ms for the 500^{th} point.

However, the choice of $\hat{\lambda}$ is not obvious: it depends on the shape of signal and the noise added. We propose to choose $\hat{\lambda} = \lambda_{ours}$, the value estimated by our algorithm or slightly larger: for the new i^{th} point, we take $\hat{\lambda}_i = \lambda_{ours}^{i-1}$ estimated from $\Lambda^{\circ,i-1}$ (the result of $i-1$ points). It is a natural choice since λ_{ours}^{i-1} needs to be calculated for restoring the $i-1$ points signal. The results of this proposition are shown in Figure 7c. The results with $\hat{\lambda} = \lambda_{ours}$ and $2\lambda_{ours}$, containing the time for estimating λ_{ours}, have a similar execution time as $\hat{\lambda} = 4$ in our simulation. Our proposition is not optimal, but it still provides a good online performance. We have to remark that the execution time is more important for each 50^{th} points since the new period of the original signal begins and the non-isolated sequence is much longer. Once again, the execution time of online implementation depends on the shape of the signal.

4.3 Real data application

The high performance of our method encourages the application in real data. We show, in Figure 8, an example of signal collected from a plant of Saint-Gobain and the restoration proposed by our approach, AUT and SURE. The original signal is unknown for the real data, so we can not compare qualitatively different methods. An alternative way is to validate the restoration by industrial process engineer. The proposition of SURE, containing some peaks, is irregular, while the solutions proposed by our approach and AUT restore well the variations of measured signal. The process engineer confirms the restoration proposed by our approach is well fitted for the further applications.

5 Conclusions and perspectives

In this paper, we analysed the behavior of Total Variation restored signal in function of the hyper-parameter λ and the introduction of a new sample at the end of the sequence. We propose different algorithms for the real time automatic TV-denoising:

- Based on [8] and [9], we propose some algorithms for estimating efficiently the restored

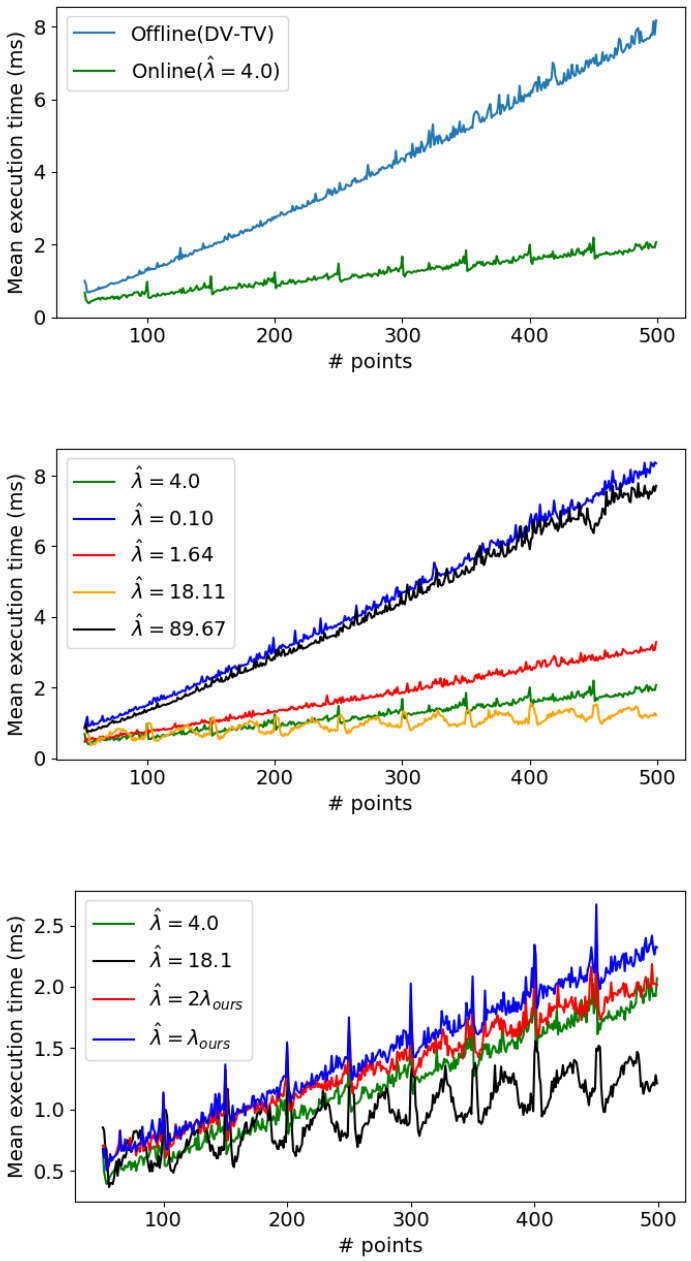


Figure 7: Mean execution time between offline and online implementation with different cutting points $\hat{\lambda}$. Execution time is averaged over 20 simulations. Top: Online vs offline (DP-TV in $O(n \log n)$); Median: Online implementation with different cutting points $\hat{\lambda}$; Bottom: A proposition of the choice of cutting point: for the i^{th} point, $\hat{\lambda}_i = \lambda_{ours}^{i-1}$ with λ_{ours}^{i-1} our hyper-parameter estimation based on the first $i-1$ points with the automatic choice of q.

signal u_{TV}^*. Our proposition combines the advantages of the existing methods: efficient for hyper-parameter selection as well as online data stream restoration.

- We propose a rapid heuristic method for selecting a good choice of λ based on the variation of the extremums number of the restored signal in function of λ. We have compared our methods with some existing methods (cross-validation, AUT and SURE) under Gaussian and/or uniform noise, and the simulations show that our method has a similar performance as cross-validation and SURE which are not compatible with the real time context. The selection of hyper-parameter stays an open question for other noise models.

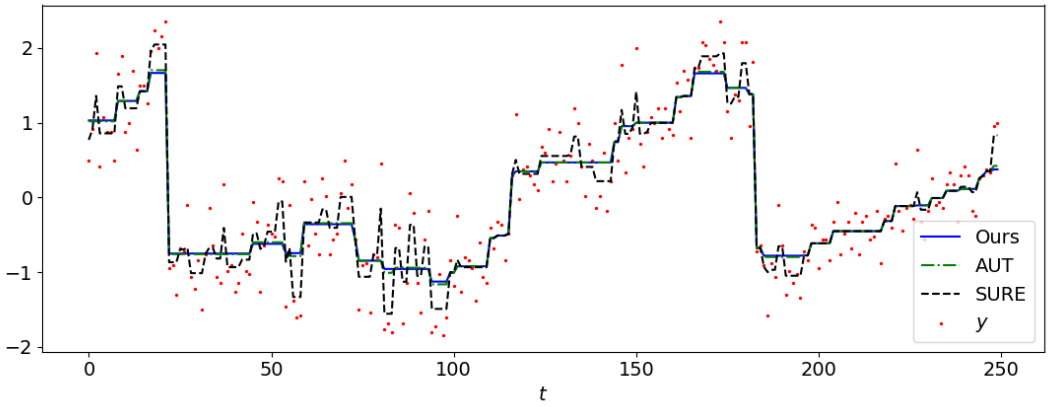


Figure 8: Real signal example: TV-denoising with our approach (blue), AUT (green) and SURE (black).

The overall complexity for obtaining a restored signal with an automatic choice of λ is in $O(n \log n)$ for the offline implementation and in $O(n)$ for the online implementation in the best case with a space complexity in $O(n)$.

The low time space complexities and the good performance of the choice of λ provide a large application field of our methods: for example, monitoring in real time a huge amount of sensors. We believe that there remains some interesting theoretical and practical issues for future research. An important theoretical and practical issue is the adaptation of the presented restoration procedure to low cost embedded devices (typically using 8 bits arithmetic).

A Proofs

We give, at first, some lemmas used by our proofs. Those lemmas are based on a fixed value of λ, so λ is omitted in most of the notation.

Lemma 1. *For the k^{th} segment $(\mathcal{N}_k^*)$ with $1 < k \leq K$, let $\mathcal{T}_{m,k} = \sum_{j=i_1^k}^{i_m^k} \tau_i$ and $\overline{y}_{m,k} = \frac{1}{\mathcal{T}_{m,k}} \sum_{j=i_1^k}^{i_m^k} \tau_j y_j$, we have the following inequalities:*

- $\overline{y}_{m,k} \geq v_k^*$ *for* $m = 1, \cdots, n_k^*$ *if* $v_{k-1}^* < v_k^*$.
- $\overline{y}_{m,k} \leq v_k^*$ *for* $m = 1, \cdots, n_K^*$ *if* $v_{k-1}^* > v_k^*$.

Lemma 2. *For the k^{th} segment $(\mathcal{N}_k^*)$ with $1 \leq k < K$: let $\mathcal{T}_{-m,k} = \sum_{j=i_{n_k^*-m}^k}^{i_{n_k}^k} \tau_i$ and $\overline{y}_{-m,k} = \frac{1}{\mathcal{T}_{-m,k}} \sum_{j=i_{n_k^*-m}^k}^{i_{n_k}^k} \tau_j y_j$, we get:*

- $\overline{y}_{-m,k} \geq v_k^*$ *for* $m = 1, \cdots, n_k$ *if* $v_{k+1}^* < v_k^*$.
- $\overline{y}_{-m,k} \leq v_k^*$ *for* $m = 1, \cdots, n_K$ *if* $v_{k+1}^* > v_k^*$.

Lemma 3.

- *If* $u_m^* = \hat{u}_m$, $\exists m \leq n$, *then* $u_i^* = \hat{u}_i$, $\forall i \leq m$.
- *If* $sign(v_{j-1}^* - v_j^*) = sign(v_j^* - \hat{v}_j)$, $\hat{v}_i = v_i^*$, $\forall i \leq j-1$.

Lemma 4. *In the case where $v^*_{K^*-1} < v^*_{K^*}$:*

- *For $y_{n+1} \geq v^*_{K^*}$, we have $v^*_j = \hat{v}_j$ and $\mathcal{N}^*_j = \hat{\mathcal{N}}_j, \forall j < K^*$.*
- *For $y_{n+1} < v^*_{K^*}$, if $\exists l$ such that $v^*_l > v^*_{l+1}$, then $v^*_j = \hat{v}_j$ and $\mathcal{N}^*_j = \hat{\mathcal{N}}_j$, $\forall j \leq l$*

Proof of Theorem 2. It is easy to show that at least one of the merged segments is a min or max segments since the neutral segment is invariant following λ.

Assume that $k+1$ segments are merged into a new segment for a given $\lambda_{l-k} \in \Lambda$. Let the new segment after merging be $\mathcal{N}_j(\lambda_{l-k}) = \{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, we will discuss the following cases:

- For $j > 1$ and $j + k < K^*(\lambda_l)$, the new segment is neither the first nor the last segment. We discuss the following cases:
  - $v^*_{j-1}(\lambda_l) > v^*_j(\lambda_l)$ and $v^*_{j+k}(\lambda_l) < v^*_{j+k+1}(\lambda_l)$: the new segment is a min segment. It can be easily shown that there must be at least one min segment among $\{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, and there must be one more min segments than max segments. So let m the number of min segments among $\{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, we have $g(\lambda_{l-k}) = g(\lambda_l) - 2(m-1)$.
  - $v^*_{j-1}(\lambda_l) < v^*_j(\lambda_l)$ and $v^*_{j+k}(\lambda_l) > v^*_{j+k+1}(\lambda_l)$: the new segment is a max segment. It can be easily shown that there must be at least one max segment among $\{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, and there must be one more max segments than min segments. So let m the number of max segments among $\{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, we have $g(\lambda_{l-k}) = g(\lambda_l) - 2(m-1)$.
  - $v^*_{j-1}(\lambda_l) > v^*_j(\lambda_l)$ and $v^*_{j+k}(\lambda_l) > v^*_{j+k+1}(\lambda_l)$ (or $v^*_{j-1}(\lambda_l) < v^*_j(\lambda_l)$ and $v^*_{j+k}(\lambda_l) < v^*_{j+k+1}(\lambda_l)$): the new segment is a neutral segment. Let m the number of min segments among $\{\mathcal{N}_j(\lambda_l), \cdots, \mathcal{N}_{j+k}(\lambda_l)\}$, we have $g(\lambda_{l-k+1}) = g(\lambda_l) - 2m$.
- For $j = 1$, the new segment is the first segment, and we discuss the following cases:
  - $v^*_{k+1}(\lambda_l) < v^*_{k+2}(\lambda_l)$: the new segment is a min segment. There must be at least one min segment among $\{\mathcal{N}_1(\lambda_l), \cdots, \mathcal{N}_{k+1}(\lambda_l)\}$, and there may be one more or equal min segments than max segments. So let m_1 the number of min segments and m_2 the number of max segments among $\{\mathcal{N}_1(\lambda_l), \cdots, \mathcal{N}_{k+1}(\lambda_l)\}$, we have $g(\lambda_{l-k}) = g(\lambda_l) - m_1 - m_2 + 1$.
  - $v^*_{k+1}(\lambda_l) > v^*_{k+2}(\lambda_l)$: the new segment is a max segment. There must be at least one min segment among $\{\mathcal{N}_1(\lambda_l), \cdots, \mathcal{N}_{k+1}(\lambda_l)\}$, and there may be one more or equal max segments than min segments. So let m_1 the number of min segments and m_2 the number of max segments among $\{\mathcal{N}_1(\lambda_l), \cdots, \mathcal{N}_{k+1}(\lambda_l)\}$, we have $g(\lambda_{l-k}) = g(\lambda_l) - m_1 - m_2 + 1$.
- For $j + k = K^*(\lambda_l)$: the new segment becomes the last one. Same as the previous points.

In summary, $g(\lambda)$ is piece-wise constant and decreasing. □

Proof of Proposition 1. Trivial from Theorem 2 □

Proof of Theorem 3. Trivial from Lemma 4. □

Proof of Theorem 4. Let $\lambda_1 < \lambda_2$, if $u^*_j(\lambda_2) \neq u^*_{j+1}(\lambda_2)$ for $j = 1, \cdots, n-1$, then $u^*_j(\lambda_1) \neq u^*_{j+1}(\lambda_1)$. If the point j is isolated from the influence of new sample for $\lambda = \lambda_2$, then it is also isolated for $\lambda = \lambda_1$. □

Proof of Proposition 2. Let l such that $\lambda_{l+1} \leq \hat{\lambda} < \lambda_l$. For $\lambda \leq \hat{\lambda}$, the merge of the elements inside j^{th} segment $(\mathcal{N}_j^*(\hat{\lambda}))$ is independent of those inside $(j-1)^{th}$ $(\mathcal{N}_{j-1}^*(\hat{\lambda}))$ and $(j+1)^{th}$ segments $(\mathcal{N}_{j+1}^*(\hat{\lambda}))$ under the same boundary conditions: $\text{sign}(u_{i_1^j} - u_{i_{n_{j-1}}^{j-1}})$ and $\text{sign}(u_{i_1^{j+1}} - u_{i_{n_j}^j})$ stay the same. The boundary conditions are guaranteed by the introduction of $-(\hat{\lambda}+\epsilon_\lambda)/(2s_{i-1}^j)$ and $(\hat{\lambda}+\epsilon_\lambda)/(2s_i^j)$ since these two new points will merge with their neighbour for $\lambda = \hat{\lambda} + \epsilon_\lambda$. So we have $\lambda^{\circ,n}_{i_1^j,\cdots,u_{i_{n_j-1}^j}} = \lambda^{*,j}_{1,\cdots,n_j-1}$.

To finish the proof, notice that $\cup\Lambda^{*,j} = \cup\lambda^{*,j}_{1,\cdots,n_j} = \cup\lambda^{\circ,n}_{i_1^j,\cdots,u_{i_{n_j-1}^j}} = \{\lambda \leq \hat{\lambda}, \forall\lambda \in \Lambda^\circ\}$. □

Proof of Lemma 1. We can easily show that $\sum_{j=i_1^k}^{i_m^k} \tau_j(y_j - \overline{y}_{m,k})^2 \leq \sum_{j=i_1^k}^{i_m^k} \tau_j(y_j - v_k^*)^2$.

For the cases in which $\overline{y}_{m,k} \in [min(v_{k-1}^*, v_k^*), max(v_{k-1}^*, v_k^*)]$, $\sum_{j=i_1^k}^{i_m^k} \tau_j(y_j - \overline{y}_{m,k})^2 + \sum_{j=i_{m+1}^k}^{i_{n_k}^k} \tau_j(y_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*) < \sum_{j\in\mathcal{N}_k} \tau_j(y_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*)$.

Since v_k^* is given by the minimisation of F_n, $\sum_{j=i_1^k}^{i_m^k} \tau_j(y_j - \overline{y}_{m,k})^2 + \sum_{j=i_{m+1}^k}^{i_{n_k}^k} \tau_j(y_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*) \geq \sum_{j\in\mathcal{N}_k} \tau_j(y_j - v_k^*)^2 + \lambda(v_k^* - v_{k-1}^*)$.

So $\overline{y}_{m,k} \notin [min(v_{k-1}^*, v_k^*), max(v_{k-1}^*, v_k^*)]$. □

Proof of Lemma 2. Same as Lemma 1. □

Proof of Lemma 3. The uniqueness of the solution of (5) is shown in [4].

$(u_1, \cdots, u_m) = \arg\min F_m$ for $m < n$ in condition of $u_m = u_m^*$. When $u_m^* = \hat{u}_m$, two solutions $(u_1^*, \cdots, u_m^*)$ and $(\hat{u}_1, \cdots, \hat{u}_m)$ minimize the same functional F_m with the same condition $u_m = u_m^*$. So $(u_1^*, \cdots, u_m^*) = (\hat{u}_1, \cdots, \hat{u}_m) = (u_1, \cdots, u_m)$.

Assuming that the first $j-1$ segments contains m points, $(v_1^*, \cdots, v_{j-1}^*) = \arg\min F_m$ under the boundary condition of $u_m^* < u_{m+1}^*$. If $\text{sign}(v_{j-1}^* - v_j^*) = \text{sign}(v_j^* - \hat{v}_j)$, the boundary condition is unchanged. So $\hat{v}_i = v_i^*$, $\forall i \leq j-1$. □

Proof of Lemma 4. The first point is trivial following Lemma 3.

For the sake of clarity, we note the last segment together with the new observation as $\mathcal{N}_a = \{\mathcal{N}_{K^*}^*, n+1\}$. For $y_{n+1} < v_{K^*}^*$, we will discuss the following cases:

- Influence on $\mathcal{N}_{K^*}^*$: we have $\hat{u}_i < u_i^*, \forall i \in \hat{\mathcal{N}_{K^*}}$ following Lemma 1.
- Influence on $\mathcal{N}_{K^*-1}^*$ with $v_{K^*-1}^* < v_{K^*}^*$: if $\hat{v}_{K^*} > v_{K^*}^*$, then $\hat{v}_{K^*-1} = v_{K^*-1}^*$ following Lemma 3. In the other case, $\{\mathcal{N}_{K^*-1}^*, \mathcal{N}_a\}$ may either merge into one segment or split into several, and $\hat{v}_{K^*-1} < v_{K^*-1}^*$ following Lemma 1. In other words, if we have a sequence $v_j^* < \cdots < v_{K^*}^*$ with $v_{i+1}^* - v_i^*$ small enough for all $i \geq j$, then y_{n+1} can change the value of the elements $\mathcal{N}_j^*$.
- Influence on $\mathcal{N}_i^*$ with $v_{i+1}^* < \cdots < v_{K^*-1}^* < v_{K^*}^*$ and $v_i^* > v_{i+1}^*$: from the previous analysis, we have $\hat{v}_{i+1} \leq v_{i+1}^*$ regardless of the value of y_{n+1} (under the condition $y_{n+1} < v_{K^*}^*$). Following Lemma 3, $\hat{v}_j = v_j^*$, and $\hat{\mathcal{N}_j} = \mathcal{N}_j^*$ for $j \leq i$.

□

Acknowledgment

The authors would like to thank David Brie (CRAN, Université de Lorraine), Paul Narchi, Diane Bienaimé (Saint-Gobain Research Paris) and Christian Fourel (Saint-Gobain Pont-à-Mousson) for their constructive advices.

This work has been supported by the EIPHI Graduate School (ANR-17-EURE-0002).

References

[1] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[2] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[4] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, 1997.

[5] Z. Harchaoui and C. Lévy-Leduc, "Multiple change-point estimation with a total variation penalty," *Journal of the American Statistical Association*, vol. 105, no. 492, pp. 1480–1493, 2010.

[6] F. Ortelli, S. van de Geer *et al.*, "On the total variation regularized estimator over a class of tree graphs," *Electronic Journal of Statistics*, vol. 12, no. 2, pp. 4517–4570, 2018.

[7] V. Kolmogorov, T. Pock, and M. Rolinek, "Total variation on a tree," *SIAM Journal on Imaging Sciences*, vol. 9, no. 2, pp. 605–636, 2016.

[8] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *The Annals of Statistics*, vol. 39, no. 3, p. 1335–1371, Jun 2011. [Online]. Available: http://dx.doi.org/10.1214/11-AOS878

[9] I. Pollak, A. S. Willsky, and Y. Huang, "Nonlinear evolution equations as fast and exact solvers of estimation problems," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 484–498, 2005.

[10] L. Condat, "A direct algorithm for 1-d total variation denoising," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1054–1057, Nov 2013.

[11] C. Louchet and L. Moisan, "Total Variation as a local filter," *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 651–694, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00457763

[12] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[13] Y.-W. Wen and R. H. Chan, "Parameter selection for total-variation-based image restoration using discrepancy principle," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1770–1781, 2011.

[14] V. A. Morozov, *Methods for solving incorrectly posed problems*. Springer Science & Business Media, 2012.

[15] S. Hashemi, S. Beheshti, R. S. Cobbold, and N. Paul, "Adaptive updating of regularization parameters," *Signal Processing*, vol. 113, pp. 228–233, 2015.

[16] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[17] J. M. Bioucas-Dias, "Bayesian wavelet-based image deconvolution: a gem algorithm exploiting a class of heavy-tailed priors," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 937–951, 2006.

[18] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[19] S. Sardy and H. Monajemi, "Efficient threshold selection for multivariate total variation denoising," *Journal of Computational and Graphical Statistics*, vol. 28, no. 1, pp. 23–35, 2019.

[20] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the l-curve," *SIAM review*, vol. 34, no. 4, pp. 561–580, 1992.

[21] A. Kempe, "Statistical analysis of discontinuous phenomena with potts functionals," Ph.D. dissertation, lmu, 2004.

[22] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476626

[23] S. Arlot, A. Celisse *et al.*, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40–79, 2010.